iTaSC as a unified framework for task specification, control, and coordination, demonstrated on the PR2

Dominick Vanthienen*, Tinne De Laet*, Wilm Decré*, Ruben Smits*, Markus Klotzbücher*, Koen Buys*, Steven Bellens*, Luca Gherardi†, Herman Bruyninckx*, and Joris De Schutter*

***Abstract*— This paper describes the implementation of the instantaneous Task Specification using Constraints (*iTaSC*)-Skill on a PR2. The framework allows easy specification and code-generation for robot tasks. Its power will be demonstrated by a mobile co-manipulation task of a human and a PR2 robot. The PR2 has to follow the instructions of the human and help him carrying an object, while avoiding obstacles and unnatural poses. Although the framework and its implementation are demonstrated on a PR2, it can be used for any robotic system. The software, including the demonstration is publicly available under an open-source license.**

Keywords: *mobile manipulation, task specification, shared control, software, PR2, iTaSC*

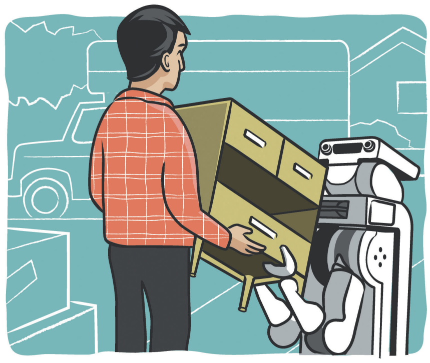

Fig. 1. Co-manipulation by a human and a robot

I. Introduction

Robots have evolved from the 6 degree-of-freedom industrial robot in a cage, to mobile platforms with redundant arms and multiple sensors, for example the PR2. These mobile platforms have escaped from their cages to a domestic, cluttered and populated environment. The resulting increase in complexity and uncertainty shows the need for a highly modular methodology for motion specification and coordination.

This paper demonstrates the capabilities of such a highly modular methodology, applied on the PR2. The demonstration includes a mobile co-manipulation task of a human and a PR2 robot, in which they handle an object together trough a cluttered and populated environment. The PR2 should follow the instructions of the human and assist him carrying the object, while avoiding obstacles. Figure 1 shows an artist's impression of the co-manipulation task.

II. iTaSC-Skills framework in a nutshell

The unified framework presented is based on ***iTaSC***, or **i**nstantaneous **Ta**sk **S**pecification using **C**onstraints, which is developed at the K.U.Leuven during the last years [3], [4], [7].

*Dept. of Mechanical Engineering, Katholieke Universiteit Leuven, Celestijnenlaan 300B, B-3001 Leuven, Belgium (email: Dominick.Vanthienen@mech.kuleuven.be). † Dept. of Information Technology and Mathematics, University of Bergamo, 24044 Dalmine, Italy

The authors gratefully acknowledge the support by Willow Garage *PR2 Beta program*, Gunter Struyf for helping to build this demo, the financial support by K.U.Leuven's GOA/2010/011, CoE EF/05/006 (*OPTEC*), the flemish *F.W.O.* project G04041N, and the European Community's Seventh Framework Programme *BRICS* FP7-231940 and *ROSETTA* FP7-230902. Tinne De Laet is a Postdoctoral Fellow of the Fund for Scientific Research–Flanders (F.W.O.) in Belgium.

The framework generates motions by specifying *constraints* in geometric, dynamic or sensor-space between the robots and their environment. These motion specifications constrain the relationships between objects (*object frames*) and their features (*feature frames*). Established robot motion specification formalisms such as the *Operational Space Approach* [5], the *Task Function Approach* [8], the *Task Frame Formalism* [6], Cartesian Space control, and Joint Space control are special cases of iTaSC and can be specified using the generic iTaSC methodology.

The key advantages of iTaSC over traditional motion specification methodologies are: (i) *composability of constraints*: multiple constraints can be combined, hence the constraints can be partial, they do not have to constrain the full 6D relation between two objects; (ii) *reusability of constraint specification*: the constraints specify a relation between feature frames, that have a semantic meaning in the context of a task, implying that the same task specification can be reused on different objects; (iii) *automatic derivation of the control solution*: the iTaSC methodology generates a robot motion that optimizes the constraints by automatically deriving the controllers from that constraint specification.

While the framework will be demonstrated on the PR2, it can be used for any robotic system, with a wide variety of sensors.

Skills are responsible for the coordinated execution of tasks and the parameter configuration of different instantaneous motion specifications. Consequently, the framework separates the continuous level motion specification and discrete level

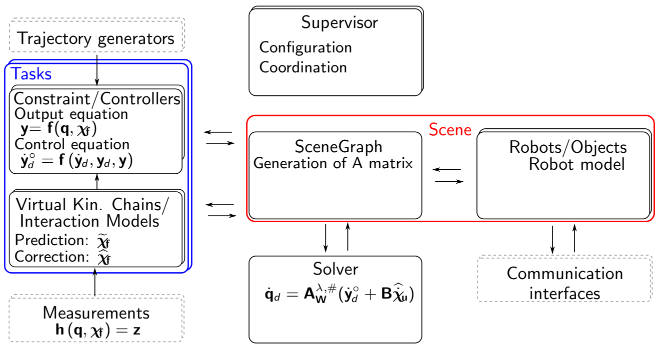

Fig. 2. Simplified software architecture scheme, with formulas for the resolved velocity case without prioritization.

coordination. One skill coordinates a limited set of constraints, that together form a functional motion. Finite State Machines implement the skill functionality.

III. Software

The *iTaSC software* implements the aformentioned framework in Orocos, which is integrated in ROS by the Orocos-ROS-integration [9]. The Real-Time Toolkit (RTT) of the Orocos project enables the control of robots on a hard-realtime capable operating system, e.g. Xenomai-Linux or RTAI-Linux [2]. The rFSM subproject of Orocos allows scripted Finite State Machines, hence Skills, to be executed in hard realtime. Figure 2 shows the software architecture, mentioning the formulas for the resolved velocity case without prioritization for clarification. The key advantages of the software design include: (i) *the modular design*, allowing users to implement their own solver, scene graph, motion generators ..., (ii) *the modular task specification* that allows users to reuse tasks, and enables a future task-web application to down- or upload tasks, (iii) *the flexible user interface*, allowing users to change the weights and priorities of different constraints, and to add or remove constraints. Furthermore, the Bayesian Filtering Library (BFL) and Kinematics and Dynamics Library (KDL) of the Orocos project are used to retrieve stable estimates out of sensor data, and to specify robot and virtual kinematic chains respectively.

IV. Demonstration

We will demonstrate a human-PR2 co-manipulation task consisting of four Skills: (i) co-manipulation of an object with the operator, (ii) dynamic and static obstacle avoidance, (iii) maintain visual contact with the operator, and (iv) unnatural pose of the PR2 prevention. In order to demonstrate the aformentioned capabilities of the iTaSC-Skill framework and software, everyone will be allowed to perform the co-manipulation with the PR2, change the weights and priorities of constraints, enable and disable constraints... A Java-based Graphical User Interface (GUI) will be placed at the operator's disposal, to interact with and monitor the state of the robot. This GUI demonstrates the possibilities of the Java-Orocos integration, JOrocos [1].

A beta version of the software, including the demonstration is publicly available under an open-source license on `http://git.mech.kuleuven.be/robotics/itasc.git`.

V. Lessons Learned

The results of the demonstration on the PR2 suggest that the performance and functionality of the real time motion control of arms and mobile base of the PR2 arms can be increased. The iTaSC-Skill software support is available under an open-source license. Its capabilities are demonstrated on a PR2, but can be used for any robotic system. We are eager to share our code and appreciate feedback on it.

Additional information including a discussion on the iTaSC–Skill software design and a teaser movie is available at `http://people.mech.kuleuven.be/~dvanthienen/IROS2011PR2/`.

References

[1] D. Brugali, L. Gherardi, M. Klotzbuecher, and H. Bruyninckx. Service component architecture in robotics: the sca-orocos integration. In *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, page to be submitted, 2011.

[2] H. Bruyninckx. Open RObot COntrol Software. `http://www.orocos.org/`, 2001. Last visited June 2011.

[3] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433–455, 2007.

[4] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter. Extending iTaSC to support inequality constraints and non-instantaneous task specification. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 964–971, Kobe, Japan, 2009.

[5] O. Khatib. The operational space formulation in robot manipulator control. In *Proceedings of the 15th International Symposium on Industrial Robots*, pages 165–172, Tokyo, Japan, 1985.

[6] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(6):418–432, 1981.

[7] J. Rutgeerts. *Constraint-based task specification and estimation for sensor-based robot tasks in the presence of geometric uncertainty*. PhD thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium, 2007.

[8] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control, the Task Function Approach*. Clarendon Press, Oxford, England, 1991.

[9] R. Smits and H. Bruyninckx. Composition of complex robot applications via data flow integration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5576–5580, Shangai, China, 2011.